A Question Detection Algorithm for Text Analysis

Tran Duc Chung
Computing Fundamental Department, FPT Technology Research Institute, FPT University, Hoa Lac Hi-Tech Park, Hanoi, Vietnam, 155300
chung.tranduc89@gmail.com
chungtd6@fe.edu.vn

Ha Hong Son
Computing Fundamental Department, FPT University, Hoa Lac Hi-Tech Park, Hanoi, Vietnam, 155300
sonhhhe140611@fpt.edu.vn

Alexandra Khalyasmaa
Automated Electrical Systems Department, Ural Federal University, Yekaterinburg, Russia, 620075
lkhalyasmaa@mail.ru

ABSTRACT
In this paper, an effective question detection algorithm for Vietnamese text analysis is proposed. The proposed algorithm takes an audio file as input, converts its speech to text, and returns question detection result. This is extremely useful for a text analyzer to determine if a given sentence generated from an audio file is a question or not, particularly in chatbot or voicebot systems where very often there are needs for automatic replies to questions queried by users. The algorithm uses two tiers of question words and a customized question phrases to achieve 88.64 % accuracy on a sub-dataset of 176 questions prepared based on FPT Open Speech Dataset.

CCS Concepts
• Information systems→Information extraction • Information systems→Sentiment analysis • Information systems→Expert search • Information systems→Data analytics • Information systems→Query intent

Keywords
Question; Detection; Speech-to-Text (STT); Text analysis; Algorithm; Chatbot; Voicebot; Application programming interface (API)

1. INTRODUCTION
In recent years, the emergence of chatbot [1, 3, 5, 7, 15] and voicebot [11, 14] systems has created a strong demand for text analysis for promptly and accurately responding to customers' inquiries. The analysis helps the bot systems to better understand a given context represented by either text or voice. Since customer services is critical to companies' businesses [6], the ultimate aim of the analysis is that the bot systems can reply to the customers' queries as correctly as possible. A fail to address customers' queries in timely manner shall result in loss of customers' interest in the products, especially in this digital world when many customers' interactions happen online and via mobile devices with limited screen view.

Figure 1 presents an example of uninteresting answers to customer's questions of an online helper system offered by a private banking corporation in Vietnam. As seen from the figure, in the first question, the answer from T'Aio [13] is totally unexpected from user point of view. Here, the customer asks to borrow money from the bank, however, the application does not support English question. In addition, it provides some alternative question phrases in Vietnamese for the customer to consider. In the second question, T'Aio provides answer that is close to customer's expectation which is a web link to view loan interest rate. However, one may see that the answer is not directly addressing the given Yes/No question. Therefore, providing correct answers to customers' queries is critically important to maintain customers' interest in using the offered application. Because the two questions and answers appearing in Figure 1 already takes up the area of half of the mobile screen, therefore, correct answers to customers' queries is essential to improve the viewing screen on customers' mobile devices.

Figure 1. Banking T'Aio [13] **chatbot's responses to customer's inquiries**

Transcript:

Customer: Can I borrow $5000 from you?

Bank (Translated): Please make very short and easy-to-understand question so that Virtual Assistant T'Aio can understand. I can answer immediately question like "find location of LiveBank", "apply for credit card", "apply for unsecured loan", …

Customer (Translated): Can I borrow $5000 from you?

Bank (Translated): Loan interest rate of TPBank is very dynamic, you can visit below link for more detail.

In [1], the concept of AI-based visual chatbot was introduced. The bot taken into consideration is a mixture of typical chatbot and

ICIIT 2020, February 19–22, 2020, Hanoi, Viet Nam

ACM ISBN 978-1-4503-7659-4/19/07…$15.00
DOI: https://doi.org/10.1145/3385209.3385230

visual content, i.e., pictures. With visual content, the conversational dialog becomes more interesting. Based on a dialog history, an image and a question about the image, the system has to answer the question correctly. This is different from voice-based chatbot system which relies on text information retrieved from recorded voices. For addressing personalized response ranking in conversation, [5] has taken into consideration of users profiles. By using contents (question-answer pairs) posted by users, their personalized representations are learnt by two neural network branches for understanding the conversation from users' perspectives. Differently, [6] utilizes user model and communication resources in developing a deep reinforcement learning network for a large financial corporation to enhance its customers experience. The main disadvantages of using AI-based systems [1, 7, 9] are long training and validation processes and requiring high performance computing cluster. In addition, for languages like Vietnamese, there are many ways to construct a question and a question can be understood based on conversational context, thus, identifying if a sentence is a question is always a challenging task.

In voice-based control system [11], often, it is required to convert voice command to text before performing control action. Understanding if a given voice is a statement or a question can help the control system to proceed with control action or reject the command respectively. Similarly, voicebot systems [4, 14, 16] also can provide responses correctly to users' queries if it can understand well whether the given queries are questions or statements. This can be achieved by processing the converted text from STT.

The key contributions of this work are: (i) a general framework for Vietnamese question detection utilizing native Python library namely Speech Recognition with STT engine from Google; and (ii) the dataset of 176 questions manually obtained from 30 hours of audio in FPT Open Speech Dataset [2].

The remaining of this paper is organized as follows: Section 2 presents research methodology; Section 3 discusses results and analysis; Section 4 concludes this research.

2. METHODOLOGY

2.1 Essential Libraries

In order to ensure effectiveness of the proposed algorithm, this work utilizes several native libraries supporting Python programming language. A brief introduction of each library is presented as follows.

- **Speech Recognition**

Speech Recognition is core library used in the proposed algorithm since it provides application programming interface (API) for recognizing speech and converting it to text [12]. There are several engines that are supported by the library:

- ✓ CMU Sphinx (works offline)
- ✓ Google Speech Recognition
- ✓ Google Cloud Speech API
- ✓ Wit.ai
- ✓ Microsoft Bing Voice Recognition
- ✓ Houndify API
- ✓ IBM Speech to Text
- ✓ Snowboy Hotword Detection (works offline)

In this work, the Google Speech Recognition engine is selected since it provides robust generated text results from various audio files compared to other engines. Furthermore, it is free of charge and supports Vietnamese language which is required in this work.

- **PyAudio and Wave**

Different from Soundfile, PyAudio [8] and Scipy [10] provide users some functions to work with *.wav audio file. The *.wav file format is selected because it is a format for lossless audio processing. In addition, this type of file is also required for the use with the Speech Recognition library.

2.2 Dataset

In this work, FPT Open Speech Dataset [2] introduced in September 2018 is utilized. The dataset comprises of totaling 30 hours of audio and its transcript. It is free and under open source license provided by FPT Vietnam aiming for academicians and research scientists to explore Vietnamese language. For the purpose of demonstration, 176 questions were obtained from the dataset and are used for testing in this work.

2.3 The Proposed Algorithm

In this work, the proposed algorithm's flowchart is presented in Figure 2. In this figure, at first, the program will import essential libraries. It then receives audio source information from user to obtain list of all audio files that need to be processed. Next, the algorithm defines key variables namely *QuestionPhrase*, *QuestionWordTier1*, *QuestionWordTier2*. The keywords stored in these variables (see Table 1) will be used for determining question given an audio file.

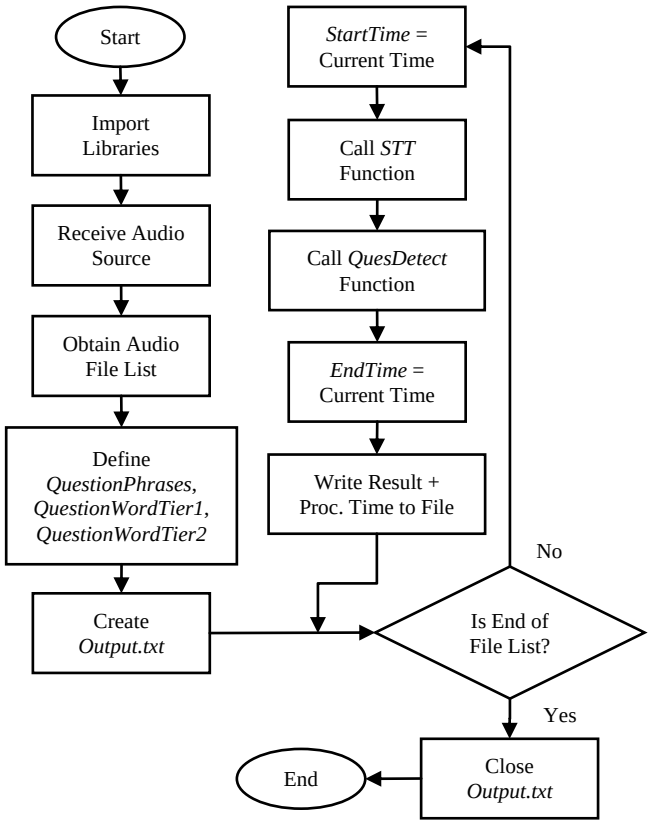

Figure 2. The proposed algorithm's flowchart

In Table 1, one can see that in Vietnamese, question phrases are very much different from English. The majority of the phrases comprise of two words while the remaining are varying from three to eight words. For question words, they are divided into two tiers: tier 1 has two words and tier 2 has only one word. When these two tiers exist in the sentence, very often the sentence is a

question. The table can be expanded to detect wider range of questions when more text is analyzed.

An *Output.txt* file is created to record processing information during execution of the algorithm. The given audio file is converted to Vietnamese text by using function *STT*. Once the text is obtained, it is detected whether a statement or question by using function *QuesDetect*. The *StartTime* and *EndTime* are noted before and after executing the above two functions respectively. The detection result and processing time (the difference between the two timings) are recorded for each processed audio file. These procedures are iterated for all audio files.

Table 1. Words / Phrase variable definitions

Variable	Words / Phrases
QuestionPhrase	"c ó kh ông", "bắt đầu l úc mấy giờ", "l àm g ì", "l à g ì", "c ó thể lấy g ì", "số mấy", "c ó ầ", "bao l âu", "c ó g ì", "khi n ào", "cho kh ông", "c ái g ì", "đi chưa", "được kh ông", "ở đâu", "khi n ào", "c ách n ào", "mấy giờ", "c ái g ì", "muốn hỏi", "vậy sao", "liệu kh ông", "rồi nhỉ", "cơ nào", "c ó ầ", "bao l âu rồi", "cho ai", "số mấy", "như nào", "thế n ào", "sao kh ông", "muốn kh ông", "c ó phải", "vậy kh ông", "đi đâu", "vội thế", "món ăn của địa phương nào đặc biệt", "ra sao", "l à sao", "thế sao", "chừng n ào", "được kh ông", "đúng không", "c ó biết", "sẵn sàng chưa", "chuyển m ấy cho ai", "anh nh ể", "mua ở đâu", "c ó n ên", "cho ph ín ào", "c òn bao nhi êu", "nghĩa là gì", "nào hơn", "hay kh ông", "mức n ào", "th ìsao", "khi n ào", "đi đâu", "c ó ở đâu", "bao nhi êu", "mấy giờ", "được kh ông", "đâu không", "là gì", "có bao giờ", "đã bao giờ", "à", "á", "đâu"
QuestionWordTier1	"có", "được", "c ó thể", "có được", "c òn"
QuestionWordTier2	"kh ông"

The *STT* algorithm is presented in Table 2. In this table, at first Speech Recognition library is imported. The Recognizer is then used for obtaining Google text from speech stored in an audio source file. Finally, the *STT* function returns the resultant text.

Table 2. *STT* algorithm

```
import speech_recognition as sr

r = sr.Recognizer()

def STT(name):
    src = sr.AudioFile(name)
    with src as source:
        audio = r.record(source)
        return r.recognize_google(audio,
language="vi-VN")
```

When detecting if a text returned by *STT* algorithm is a question or a statement, *QuesDetect* function described in Table 3 is used. From the table, it is seen that the function takes four parameters: the given text, question phrase and two-tier question words. Before processing the text, it is lowercased to normalize all characters. The function will immediately categorize the text as a question if a question phrase defined in Table 1 is found in the text. In another case, when both question word tiers appeared in the text and in ascending order, the text will be categorized as a question. In the remaining cases, the text will be categorized as a statement.

Table 3. Question detection algorithm

```
def QuesDetect(text, QuestionPhrase,
QuestWordTier1, QuestWordTier2) :
    for i in QuestionPhrase:
        if(i in text.lower()):
            return True

    for i in QuestWordTier1:
        if(i in text.lower()):
            for j in QuestWordTier2:
                if j in text and
text.index(j) > text.index(i):
                    return True
    return False
```

3. RESULTS & ANALYSIS

When testing with a dataset comprising of 176 audio files manually obtained from FPT Open Speech Dataset, the correctly detected questions is 156, thus the algorithm's accuracy is approximately 88.64 %. Further analyzing the false to detect question cases, Table 4 presents two categories of wrong detection cases namely missing word and wrong context.

Table 4. Classification of wrong detection cases

No.	Name	Quantity
1	Missing Word	18
2	Wrong Context	2

As seen from the above table, the majority of the wrong detection cases fall under the first category which depends on several factors, to name a few: STT engine, audio quality, audio noise, Internet connection quality between local host and Google server, etc. It is expected that if there are less noise, the signal quality is better, the algorithm shall achieve higher accuracy. Therefore, the audio source should be pre-handled for reducing noise level before converting to text.

An example of missing word category is presented in Figure 3. Here, the audio file has *.wav format with two channels and sampling rate of 44,100 Hz which is the same as in the other audio files. The file has duration of approximately 1.5 s and size of 260 kb. This information can be obtained from PyAudio and Scipy libraries.

```
This is audio file number 2
File's Name:          : FPTOpenSpeechData_Set001_V0.1_000007
File's Type:          : WAV
Channel:              : 2
Sample rate           : 44100
Duration              : 0:00:01.512018
File's size:          : 260 KB
Content: Anh đổi ngoại tệ được
This is a statement
Processing time: 0.497023 sec
```

Figure 3. Example for word-missing mistake

In the figure, the STT Google API returned a text with some missing words at the end of the text which is usually the question

word, i.e., "không" ("no" in English). In this case, the retrieved content "Anh đổi ngoại tệ được" ("He can exchange foreign currency.") indicates this is a statement while the original audio content: "Anh đổi ngoại tệ được không" ("Can he exchange foreign currency?") represents a question instead. This means that the question detection results' accuracy relies strongly on the accuracy of STT engine.

An example of wrong context category is presented in Figure 4. This type of wrong detection is very difficult to be corrected since the question meaning depends on situation and the rhythms of the recorded speech in the audio file. In Figure 4a, the returned string is "*bạn có quan tâm đến đồ cũ*" ("You are interested in old stuff") while the original speech is "*bạn có quan tâm đến đồ **cổ không***" ("Are you interested in old stuff?"). The last word is missing from the returned text and in addition, the word "cổ" ("antique") has been changed to "cũ" ("old"). Although they have same meaning, however, given the returned text, one can either understand it is a question or a statement depending on the context. In Figure 4b, the returned text ("*quý vị có thể cử động khi đầy đủ*" - "you can move when it is full") has totally different meaning from the original speech ("*quý vị có thể cử động mắt cá chân được không*" - "can you move your ankle?").

```
Content: bạn có quan tâm đến đồ cũ
This is a statement
Processing time: 1.25125 sec
```

a)

```
Content: quý vị có thể cử động khi đầy đủ
This is a statement
Processing time: 2.078686 sec
```

b)

Figure 4. Example of wrong context

Table 5 presents processing performance of the proposed algorithm. The total processing time for the aforementioned question set is approximately 160 s. On average, each audio file takes 0.907 s to be processed. The standard deviation is 0.458 s which is about half of the mean processing time. This is mainly due to differences in lengths of the audio files. At most, it takes 2.593 s and at least it takes 0.344 s to process the audio files. In total, there are 5,956 Unicode characters generated from the STT, thus, on average it takes 0.02681 s to generate a character from speech.

Table 5. Processing performance

No.	Parameter		Quantity
1	Processing Time (s)	Total	159.707
		Mean	0.907
		Standard Deviation	0.458
		Max	2.593
		Min	0.344
2	Total Generated Characters		5956
3	Min Character		8
4	Max Character		81

In general, it is difficult to detect if a speech is a question or statement when the returned text from STT is missing an important question word, some words or has totally different meaning (wrong context).

For applying to other languages like English, etc., one may reconfigure the Google API if it does support conversion of speech to text in such language. In addition, one may further add question words into the algorithm and its question detection capability will be improved.

4. CONCLUSION

In conclusion, this work has presented a framework to detect questions from audio files which is applicable to Vietnamese language, extensible to other languages. The algorithm was tested on samples taken from FPT Open Speech Data and it is evident that the algorithm can achieve 88.64 % accuracy. Future works will further improve the algorithm's accuracy taking into consideration of sentimental analysis of the questions.

5. ACKNOWLEDGMENTS

The authors would thank FPT University and Ural Federal University for supporting this research. In addition, the authors would thank the students: Nguyen Khuong Quan, Tran Viet Thai, Le Sy Thanh Long in SE1402 class, FPT University for their partial support this research.

6. REFERENCES

[1] Das, A. et al. 2019. Visual Dialog. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. (2019). DOI:https://doi.org/10.1109/TPAMI.2018.2828437.

[2] FPT Open Speech Data: 2018. *https://fpt.ai/fpt-open-speech-data/*.

[3] Gent, E. 2014. Imitation brains. *Engineering and Technology*. (2014). DOI:https://doi.org/10.1049/et.2014.1101.

[4] Kepuska, V. and Bohouta, G. 2018. Next-generation of virtual personal assistants (Microsoft Cortana, Apple Siri, Amazon Alexa and Google Home). *2018 IEEE 8th Annual Computing and Communication Workshop and Conference, CCWC 2018* (2018).

[5] Liu, B. et al. 2018. Content-Oriented User Modeling for Personalized Response Ranking in Chatbots. *IEEE/ACM Transactions on Audio Speech and Language Processing*. (2018). DOI:https://doi.org/10.1109/TASLP.2017.2763243.

[6] Liu, Z. et al. 2019. Which Channel to Ask My Question?: Personalized Customer Service Request Stream Routing Using Deep Reinforcement Learning. *IEEE Access*. (2019). DOI:https://doi.org/10.1109/ACCESS.2019.2932047.

[7] Mao, G. et al. 2019. Multi-Turn Response Selection for Chatbots With Hierarchical Aggregation Network of Multi-Representation. *IEEE Access*. (2019). DOI:https://doi.org/10.1109/access.2019.2934149.

[8] PyAudio: 2019. *https://pypi.org/project/PyAudio/*.

[9] Qayyum, A. et al. 2019. Convolutional Neural Network Approach for Estimating Physiological States Involving Face Analytics. *2019 IEEE International Conference on Automatic Control and Intelligent Systems, I2CACIS 2019 - Proceedings* (2019).

[10] SciPy: 2019. *https://pypi.org/project/scipy/*.

[11] Solorio, J.A. et al. 2018. Voice Activated Semi-Autonomous Vehicle Using off the Shelf Home Automation Hardware. *IEEE Internet of Things Journal*. (2018). DOI:https://doi.org/10.1109/JIOT.2018.2854591.

[12] SpeechRecognition: 2019. *https://pypi.org/project/SpeechRecognition/*.

[13] TPBank: 2019. *https://tpb.vn/*.

[14] Tulshan, A.S. and Dhage, S.N. 2019. Survey on virtual assistant: Google assistant, Siri, Cortana, Alexa. *Communications in Computer and Information Science* (2019).

[15] Wang, Y. et al. 2019. Augmenting Dialogue Response Generation with Unstructured Textual Knowledge. *IEEE Access*. (2019). DOI:https://doi.org/10.1109/ACCESS.2019.2904603.

[16] Yuan, X. et al. 2018. All Your Alexa Are Belong to Us: A Remote Voice Control Attack against Echo. *2018 IEEE Global Communications Conference, GLOBECOM 2018 - Proceedings* (2018).